

WHAT IS CLAIMED IS:

1. A computer-implemented method for obtaining scratch registers for use by a computer-executable binary, comprising:
 - (a) determining a maximum number of registers requested from a plurality of register requests; and
 - (b) modifying each register request in the plurality of register requests to request the maximum number of registers plus an additional number of registers.
2. The method of claim 1, wherein the additional number of registers corresponds to a selected number of scratch registers.
3. The method of claim 1, wherein a procedure in the computer-executable binary includes the plurality of register requests.
4. The method of claim 3, wherein the additional number of registers corresponds to a selected number of scratch registers and wherein each of the scratch registers is indexed by an index that remains constant throughout the procedure.
5. The method of claim 1, further comprising
 - (c) using at least one of the plurality of modified register requests to support instrumentation code in the computer-executable binary.
6. The method of claim 1, wherein the computer-executable binary is constructed for execution on a processor configured to execute a speculative instruction.
7. The method of claim 6, wherein data is stored in a register in association with the speculative instruction, and wherein moving the data to a main memory results in a hardware fault.

8. A computer system, comprising:
- (a) a computer-executable binary;
 - (b) a procedure boundary detector configured to identify a procedure of the computer-executable binary; and
 - (c) a scratch register allocator configured to receive the identified procedure from the procedure boundary detector and to modify the computer-executable binary to request scratch registers.
9. The system of claim 8, wherein the computer-executable binary comprises at least one register allocation request.
10. The system of claim 9, wherein the scratch register allocator provides at least one scratch register by modifying the at least one register allocation request.
11. The system of claim 8, further comprising:
- (d) a basic block detector configured to receive the identified procedure from the procedure boundary detector and to identify at least one basic block in the identified procedure; and
 - (e) a dominating register allocation detector configured to receive the at least one basic block and to detect at least one dominating allocation for the at least one basic block
- wherein the scratch register allocator is further configured to receive the at least one basic block identified and the at least one dominating allocation detected.
12. The system of claim 11, wherein the basic block detector is further configured to construct a control flow graph using the at least one basic block identified.
13. The system of claim 12, wherein the scratch register allocator is further configured to receive the control flow graph.

14. A computer-readable medium having computer-executable instructions, comprising:
- (a) discovering a procedure in a computer-executable instructions;
- and
- (b) if a register allocation does not exist at the beginning of the procedure, inserting a register allocation.
15. The method of claim 14, further comprising:
- (c) determining a maximum number of registers requested in the procedure; and
 - (d) modifying each register request in the procedure to request the maximum number of registers requested plus a number of scratch registers.
16. The method of claim 14, further comprising:
- (c) for a basic block in the procedure:
 - (i) finding at least one dominating allocation;
 - (ii) modifying the at least one dominating allocation to request a number of scratch registers.
17. The method of claim 14, further comprising:
- (c) finding at least one basic block in the procedure;
 - (d) constructing a control flow graph from the at least one basic block;
 - (e) using the control flow graph to discover at least one dominating allocation; and
 - (f) modifying the at least one dominating allocation to request a number of scratch registers.